Dynamic Hybrid Forecasting Models for Drug Consumption Prediction in Hospital Pharmacies

Yuxin Fan[*] and Siye Wu[†]
[*]School of Engineering and Applied Science, University of Pennsylvania, Canada, Toronto
Email: yuxinfan@alumni.upenn.edu
[†]Simon Business School, University of Rochester, Canada, Toronto
Email: april.siyewu@hotmail.com

Abstract—**Accurate and efficient drug consumption forecasting is crucial for hospital pharmacies to avoid overstocking, minimize wastage, and ensure continuous patient care. This study proposes a hybrid forecasting framework integrating XGBoost, Prophet, and SARIMAX to improve monthly consumption predictions at the drug-manufacturer level. Through rolling-window forecasting and advanced feature engineering, the proposed approach addresses challenges such as seasonality, trend shifts, and sparse data. Experimental results demonstrate significant improvements in prediction accuracy and robustness across diverse drug consumption scenarios.**

Index Terms—**Drug Consumption Forecasting, Hospital Pharmacies, Forecasting Models, XGBoost, Prophet, SARIMAX**

I. Introduction

Drug consumption forecasting plays a critical role in hospital pharmacy inventory management. Accurate predictions enable hospital pharmacies to ensure drug availability while minimizing costs associated with overstocking or stockouts. As highlighted by Koala et al. [1], forecasting drug consumption is particularly challenging due to numerous influencing factors, such as sociodemographic characteristics, morbidity patterns, drug price index, and seasonal factors like disease outbreaks or policy changes. This study aims to develop a hybrid forecasting framework tailored to address these challenges by integrating XGBoost, Prophet, and SARIMAX with rolling-window forecasting and advanced feature engineering.

In previous studies, various forecasting techniques have been proposed to address one or more of these complexities. Taylor and Letham [2] introduce Prophet, a scalable forecasting method designed for large-scale applications, focusing on capturing general trends and seasonality with changepoint detection. However, Prophet is limited in handling external variables and often requires careful tuning to avoid overfitting to local patterns. Machine learning techniques, such as XGBoost proposed by Chen and Guestrin [3], effectively model non-linear relationships but struggle with sequential dependencies in time-series data.

Furthermore, comparative and hybrid studies like those by Ferreira et al. [4] and Meng et al. [5] highlight that while LSTM improves non-linear pattern recognition compared to ARIMA and Prophet excels in handling sparse data, both face challenges with computational complexity and adaptability. Xu et al. [6] demonstrate a hybrid approach combining linear regression and LSTM networks, which improves temporal forecasting but struggles with capturing complex, non-linear interactions across diverse datasets. Siddiqui et al. [7] explore hybrid models like ARIMA-Holt's Winter, which improve upon traditional statistical methods but rely heavily on fixed architectures, limiting adaptability in dynamic environments. Rathipriya et al. [8] utilize hybrid neural networks to address temporal patterns but face challenges with data sparsity and computational efficiency.

In this study, to address the above limitations, our proposed framework introduces rolling-window forecasting to adjust predictions for non-stationary data and shifting trends dynamically, along with advanced feature engineering that enriches the data with lag features and trend indicators, capturing both short-term dependencies and long-term dynamics. These techniques are seamlessly integrated with hybrid complementary models: XGBoost, which models non-linear relationships and feature interactions; SARIMAX, which incorporates external variables and seasonality; and Prophet, which handles trend decomposition and missing data. This synergy provides a robust and versatile solution to the unique challenges of drug consumption forecasting, making the framework applicable across various drug types and manufacturers.

The remainder of this paper is organized as follows: Section II details the methodology, including data preprocessing and model design. Section III presents experimental results and discussions, highlighting the advantages of the proposed framework. Finally, Section IV concludes the study and outlines future research directions.

II. Methodology

A. Introduction to Hybrid Model Framework

The proposed hybrid forecasting framework integrates the complementary strengths of Prophet, SARIMAX, and XGBoost to address the challenges of drug consumption prediction. Unlike single-model approaches that are limited in their ability to generalize across diverse data characteristics, this framework leverages the distinct strengths of each model to provide more comprehensive and accurate predictions. By dynamically selecting the most appropriate model for each drug-manufacturer combination, the framework ensures that varying data patterns, including seasonal variations, non-linear dependencies, and sparse observations, are effectively captured.

A key challenge in drug consumption prediction is the variability in data characteristics across different drug-manufacturer combinations. Some datasets exhibit consistent and stable trends over time, while others may show significant volatility or sparsity. For example, drugs with high and regular demand may be well-suited to models like Prophet, which excels at decomposing long-term trends and seasonal components. Conversely, drugs with irregular consumption patterns or strong external influences may be better explained by SARIMAX, which incorporates external variables and captures short-term dynamics through advanced feature engineering. In cases where non-linear interactions and complex feature dependencies dominate, XGBoost is better equipped to model these relationships due to its gradient boosting mechanism and flexibility in feature selection.

To account for these differences, the framework evaluates each model's performance on specific drug-manufacturer combinations using metrics such as R^2 and symmetric mean absolute percentage error (SMAPE). These metrics provide a quantitative basis for selecting the model that best explains the underlying data patterns. For instance, a high R^2 value indicates that a model effectively captures the variance in the data, while a low SMAPE suggests robust performance in capturing relative changes across time. By dynamically assigning the most suitable model to each combination, the framework maximizes prediction accuracy and minimizes the limitations associated with single-model approaches.

This model selection strategy highlights the heterogeneity in drug consumption trends and emphasizes the need for a hybrid approach. Drugs with stable and predictable behavior are efficiently handled by models like Prophet, while SARIMAX addresses datasets influenced by external factors or exhibiting strong seasonality. XGBoost complements these statistical models by providing superior accuracy for datasets with non-linear relationships or sparse observations. Through this collaborative mechanism, the hybrid framework not only enhances prediction robustness but also expands the scope of application to a wider variety of drug consumption scenarios.

B. Model Design and Roles

The hybrid forecasting framework integrates Prophet, SARIMAX, and XGBoost to leverage their respective strengths in addressing the challenges of drug consumption prediction. Each model is designed to fulfill a specific role, complementing the others to achieve higher accuracy and robustness. Prophet focuses on capturing long-term trends and seasonality, SARIMAX incorporates external variables and advanced feature engineering for short-term dependencies, and XGBoost models complex non-linear interactions among features. Together, they form a cohesive framework for dynamic and adaptive time-series forecasting.

1) SARIMAX Model: SARIMAX extends the traditional ARIMA model by incorporating external (exogenous) variables, allowing it to capture external influences on the target variable. This capability makes SARIMAX particularly suitable for time-series data with seasonality, trends, and additional contextual factors. The model is defined as:

$$y_t = \phi(B)\theta(B)^{-1}\left(c + \mathbf{X}_t\beta + \epsilon_t\right), \tag{1}$$

where:

- y_t: The target variable (e.g., monthly drug consumption).
- $\phi(B)$: The autoregressive (AR) operator.
- $\theta(B)$: The moving average (MA) operator.
- c: A constant term.
- $\mathbf{X}_t$: A vector of exogenous variables at time t.
- β: The coefficient vector for $\mathbf{X}_t$.
- ϵ_t: White noise error term.

To enhance prediction accuracy, SARIMAX incorporates various exogenous variables derived through advanced feature engineering. These include:

Lagged values: Lagged values capture the delayed effects of past consumption and are defined as:

$$\mathrm{lag}_k = y_{t-k}, \quad k \in \{1, 3, 6, 12\}. \tag{2}$$

Rolling statistics: Rolling statistics represent short-term trends and variability. They include the rolling mean and standard deviation:

$$\text{Rolling Mean}_k = \frac{1}{k}\sum_{i=1}^{k} y_{t-i}, \tag{3}$$

$$\text{Rolling Std}_k = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(y_{t-i} - \text{Mean})^2}. \tag{4}$$

Exponential weighted moving average (EWMA): EWMA emphasizes recent observations with a smoothing factor α:

$$\mathrm{EWMA}_\alpha = \alpha y_t + (1-\alpha)\cdot \mathrm{EWMA}_{\alpha,t-1}. \tag{5}$$

Seasonality variables: Monthly seasonality is encoded using trigonometric functions:

$$\text{Month\_sin} = \sin\left(\frac{2\pi \cdot \text{Month}}{12}\right), \tag{6}$$

$$\text{Month\_cos} = \cos\left(\frac{2\pi \cdot \text{Month}}{12}\right). \tag{7}$$

Percentage change: Percentage change measures relative variations over time:

$$\text{Pet\_Change}_1 = \frac{y_t - y_{t-1}}{y_{t-1}}, \tag{8}$$

$$\text{Pet\_Change}_3 = \frac{y_t - y_{t-3}}{y_{t-3}}. \tag{9}$$

Trend and volatility: Trend strength captures long-term dynamics, while volatility measures variability:

$$\text{Trend Strength} = \frac{1}{k}\sum_{i=1}^{k}|y_{t-i} - y_{t-i-1}|, \tag{10}$$

$$\text{Volatility} = \text{Rolling Std}_k. \tag{11}$$

These exogenous variables, combined with SARIMAX's inherent ability to model seasonality and trends, enable it to effectively capture complex temporal dependencies and improve prediction accuracy.

2) XGBoost Model: XGBoost is a gradient boosting framework designed to construct an ensemble of decision trees for predicting target variables. Its ability to capture non-linear relationships and complex interactions among features makes it highly effective for time-series forecasting tasks, particularly in scenarios involving dynamic patterns and sparse data. The model predicts the target variable y_t through an additive function:

$$\hat{y}_t = F(x_t) = \sum_{k=1}^{K} f_k(x_t), \quad f_k \in \mathcal{F}, \tag{12}$$

where $\hat{y}_t$ is the predicted value at time t, x_t represents the input features, f_k denotes the k-th decision tree, and $\mathcal{F}$ is the function space of decision trees.

The predictive power of XGBoost in this framework is enhanced through careful feature engineering. Temporal dependencies are captured by incorporating lagged values, such as historical consumption data $(y_{t-1}, y_{t-2}, \ldots, y_{t-k})$, which model delayed effects of past consumption. Rolling statistics, including moving averages and standard deviations over defined windows (e.g., 3, 6, and 12 periods), are introduced to quantify local trends and variability. Seasonal patterns are encoded using trigonometric functions ($\sin$ and $\cos$) to capture monthly or quarterly cycles, and interaction terms are derived to represent the interplay between lagged values and seasonal indicators. This comprehensive feature set enables XGBoost to model both short-term fluctuations and long-term patterns effectively.

To address the non-stationarity inherent in time-series data, the model is trained iteratively using a rolling-window approach. At each iteration, the training dataset is updated to include the most recent observations while maintaining a fixed window size. The model is then retrained on this updated dataset, and predictions are made for the subsequent time step. This rolling-window strategy ensures that the model dynamically adapts to evolving trends and minimizes the risk of overfitting to older data.

Hyperparameter optimization plays a crucial role in maximizing the model's performance. During each training iteration, a grid search is conducted to tune parameters such as the number of estimators, learning rate, maximum tree depth, subsample ratio, and column sampling ratio. The number of estimators determines the size of the ensemble, while the learning rate controls the contribution of each tree to the final prediction. Maximum tree depth limits the complexity of individual trees to prevent overfitting, and the subsample and column sampling ratios regulate the fraction of data and features used for training, enhancing the model's generalization capability.

XGBoost serves as a critical component of the hybrid forecasting framework by complementing the strengths of statistical models like SARIMAX. Its ability to model non-linear interactions and complex dependencies allows it to address challenges posed by irregular patterns and sparse data. Moreover, its integration with rolling-window training and advanced feature engineering ensures robustness and adaptability, making it a valuable tool for predicting drug consumption trends in diverse scenarios.

3) Prophet Model: Prophet is a robust time-series forecasting model developed by Facebook, designed to explicitly decompose time-series data into trend, seasonality, and holiday components. Its ability to handle missing values, outliers, and irregular patterns makes it particularly suitable for real-world applications involving complex and dynamic data. The model predicts the target variable y_t as:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t, \tag{13}$$

where $g(t)$ represents the long-term trend, $s(t)$ models recurring seasonal patterns using Fourier series, $h(t)$ accounts for holiday effects or special events, and ϵ_t is a white noise error term. This decomposition framework allows Prophet to capture distinct components of time-series data independently, providing interpretable results while maintaining high predictive accuracy.

To ensure the model's adaptability to different drug-manufacturer combinations, key hyperparameters are optimized through grid search. These parameters include: - *seasonality_mode*, which determines whether seasonal patterns are modeled as additive or multiplicative effects; - *changepoint_prior_scale*, controlling the flexibility of the trend component by determining the likelihood of abrupt changes in growth; - *seasonality_prior_scale*, which adjusts the weight assigned to seasonal components.

The grid search systematically explores combinations of these hyperparameters, enabling the model to identify configurations that best capture the temporal dynamics of drug consumption. For example, a higher *changepoint_prior_scale* allows the model to adapt to datasets with frequent structural changes in trend, while a lower value favors smoother transitions.

Prophet's rolling-window forecasting strategy further enhances its robustness. By iteratively retraining the model on the most recent data, the framework dynamically incorporates emerging trends and minimizes the impact of outdated patterns. This approach, combined with the model's inherent decomposition capabilities, ensures accurate and interpretable predictions, even for datasets with irregular or sparse observations.

C. Dynamic Rolling-Window Forecasting

Time-series data often exhibit non-stationarity, where patterns such as trends, seasonality, and noise evolve over time. Static forecasting approaches, which rely on fixed historical data, may fail to adequately capture these dynamics, leading to suboptimal performance. To address this challenge, a dynamic rolling-window forecasting strategy is employed in this study, enabling the models to adaptively update their predictions by leveraging the most recent information.

The rolling-window mechanism operates iteratively. At each time step t, the training dataset is updated to include the most recent observations while discarding older data beyond the defined window size. This dynamic adjustment ensures that the models prioritize recent patterns, which are often more predictive of future behavior, while mitigating the influence of outdated or less relevant data. Formally, the training dataset at time t can be expressed as:

$$\mathcal{D}_t = \{(y_\tau, \mathbf{X}_\tau) \mid \tau \in [t-W, t-1]\},$$

where $\mathcal{D}_t$ represents the dataset used for training at time t, W is the window size, y_τ denotes the target variable, and $\mathbf{X}_\tau$ encompasses the corresponding feature vectors. After training on $\mathcal{D}_t$, predictions are generated for the next time step $(t+1)$.

This strategy is particularly advantageous for capturing short-term dynamics in non-stationary data. By continuously updating the training window, the models are able to respond to structural changes in the data, such as shifts in trends or seasonal patterns. Furthermore, the rolling-window approach inherently supports the detection of emerging behaviors, ensuring that the forecasting framework remains robust in dynamic and evolving environments.

A critical consideration in rolling-window forecasting is the determination of an appropriate window size (W). A larger W incorporates long-term historical information, which may benefit datasets with persistent trends or pronounced seasonality. Conversely, a smaller W focuses on recent observations, offering greater sensitivity to abrupt changes or irregular patterns. To balance these trade-offs, this study empirically evaluates multiple window sizes for each model and selects the optimal configuration based on performance metrics such as root mean squared error (RMSE) and symmetric mean absolute percentage error (SMAPE).

The integration of rolling-window forecasting into the hybrid framework further enhances its adaptability. For example, SARIMAX dynamically recalibrates its coefficients using the updated dataset, allowing it to effectively model short-term dependencies and external influences. Similarly, XGBoost leverages the rolling-window mechanism to refine its decision trees, incorporating the most recent feature interactions to capture evolving non-linear relationships. Prophet, with its inherent trend decomposition capabilities, also benefits from the dynamic updating of its training set, ensuring that its predictions remain aligned with the latest trends.

In summary, the dynamic rolling-window forecasting strategy underpins the robustness and flexibility of the proposed framework. By iteratively incorporating recent information while adapting to structural changes, this approach ensures that the models remain both responsive and resilient, achieving superior forecasting performance across diverse drug consumption scenarios.

D. Implementation and Optimization

The implementation of the proposed forecasting framework emphasizes modularity, scalability, and computational efficiency. Each model is designed to operate within a unified pipeline that supports dynamic updates, enabling seamless integration of rolling-window forecasting and hyperparameter optimization.

The forecasting process is organized into three primary stages: data preprocessing, model training, and evaluation. In the preprocessing stage, raw data is transformed into feature-rich datasets through advanced feature engineering, ensuring compatibility with the unique requirements of each model. The training stage employs dynamic rolling-window strategies, allowing each model to adapt to the most recent data while leveraging grid search to optimize key hyperparameters. Finally, the evaluation stage utilizes metrics such as root mean squared error (RMSE) and symmetric mean absolute percentage error (SMAPE) to assess forecasting performance across diverse drug-manufacturer combinations.

To ensure computational efficiency, the framework incorporates parallel processing and model-specific optimizations. For example, XGBoost utilizes multi-threading capabilities to accelerate decision tree construction, while Prophet leverages Fourier series approximations to efficiently model seasonal components. The SARIMAX implementation is enhanced by efficient matrix operations for parameter estimation, particularly when handling large datasets with multiple external variables.

The framework's modular design allows for seamless adaptation to new data sources and forecasting scenarios. Detailed implementation details, including pseudocode and algorithmic workflows, are provided in Appendix A to facilitate reproducibility and scalability for future applications.

E. Dynamic Feature Engineering

Feature engineering is crucial for enhancing model performance. Besides the exogenous variables described above, outlier detection is a key preprocessing step.

1) Outlier Detection and Handling: Outliers can distort predictions and degrade model performance. This study employs two approaches for outlier detection:

- **Z-score Method**: The Z-score for each data point y_i is calculated as:
$$Z_i = \frac{y_i - \mu}{\sigma}, \tag{14}$$
where μ is the mean and σ is the standard deviation of the dataset. Data points with $|Z_i| > 3$ are identified as outliers.
- **Interquartile Range (IQR) Method**: The IQR is defined as:
$$\text{IQR} = Q_3 - Q_1, \tag{15}$$
where Q_1 and Q_3 are the 25th and 75th percentiles, respectively. A data point y_i is considered an outlier if:

$$y_i < Q_1 - 1.5 \cdot \text{IQR} \quad \text{or} \quad y_i > Q_3 + 1.5 \cdot \text{IQR}. \tag{16}$$

To mitigate the effect of outliers, values exceeding these thresholds are capped at the 5th and 95th percentiles of the data distribution:

$$y_i = \begin{cases} \text{Percentiles}, & \text{if } y_i < \text{Percentiles} \\ \text{Percentile}_{95}, & \text{if } y_i > \text{Percentile}_{95}. \end{cases} \tag{17}$$

2) Rolling-Window Forecasting: To adapt to dynamic consumption patterns, a rolling-window mechanism is implemented. At each prediction step t, the model is trained using historical data $\{y_1, y_2, \ldots, y_t\}$, and the prediction for $t+1$ is made. The window then updates to include the latest observation, ensuring that the model adapts to recent trends.

III. Experiments

A. Experimental Setup

The dataset consists of monthly drug consumption records, including:

- Drug Name
- Manufacturer
- Monthly Consumption

Models are evaluated using the following metrics:

1) **Root Mean Squared Error (RMSE)**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{18}$$

2) **Mean Absolute Error (MAE)**:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|. \tag{19}$$

3) **Symmetric Mean Absolute Percentage Error (SMAPE)**:

$$\text{SMAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \times 100\%. \tag{20}$$

B. Results

The performance of the models is summarized in Table I.

IV. Discussion and Future Work

A. Model Comparison and Hybrid Framework

While each model has unique strengths and limitations, integrating their complementary capabilities within a hybrid framework provides significant advantages:

- **XGBoost** handles nonlinear relationships effectively.
- **SARIMAX** captures seasonality and leverages external variables.
- **Prophet** provides robust decomposition of trend and seasonality.

The hybrid approach enables the framework to adapt to different data characteristics, achieving better overall prediction performance. For example, SARIMAX improves accuracy when seasonality dominates, whereas XGBoost captures complex interactions in data with dynamic patterns.

B. Future Work

To further improve drug inventory predictions, the following directions are proposed for future research:

- **Hybrid Framework Expansion**: Extend the hybrid framework by integrating additional models, such as LightGBM, Transformer-based architectures, or ensemble methods to capture both short-term fluctuations and long-term trends.
- **External Variable Enrichment**: Explore additional exogenous variables such as patient inflow, epidemic outbreak data, seasonal illnesses (e.g., flu seasons), or hospital-specific factors to improve prediction accuracy.
- **Automated Model Selection**: Implement automated hyperparameter tuning and model selection techniques (e.g., Bayesian optimization) to improve forecasting performance with minimal manual intervention.

By addressing these areas, the proposed framework can become a more robust and scalable solution for hospital pharmacy inventory management, ensuring continuous patient care and reducing operational inefficiencies.

V. Conclusion

This study evaluated the performance of XGBoost, SARIMAX, and Prophet for drug inventory prediction in hospital pharmacies. The main findings are as follows:

- XGBoost outperformed SARIMAX and Prophet in terms of RMSE and SMAPE for short-term predictions.
- SARIMAX demonstrated better performance when external (exogenous) variables were included and seasonal trends dominated the data.
- Prophet performed well in capturing long-term seasonality and trends but showed limitations in handling highly sparse data.

The hybrid framework, by combining these models, shows promise for improving prediction accuracy and robustness. Future work will focus on integrating real-time data streams and exploring additional hybrid models, such as Transformer-based architectures, to further enhance predictive capabilities.

VI. References

References

[1] D. Koala, Z. Yahouni, G. Alpan, and Y. Frein, “Factors influencing drug consumption and prediction methods,” in *CIGI-Qualita: Conférence Internationale Génie Industriel QUALITA*, Grenoble, France, 2021.

[2] S. J. Taylor and B. Letham, “Forecasting at scale,” *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[4] D. Ferreira, P. Teixeira, and A. Dias, “Exploring the performance of ARIMA and LSTM in time series forecasting: A comparative study,” *International Journal of Computer Science and Applications*, vol. 15, no. 2, pp. 20–34, 2018.

[5] J. Meng, Q. Zhang, and X. Li, “Comparative analysis of Prophet and LSTM models in drug sales forecasting,” *Journal of Physics: Conference Series*, vol. 1910, no. 1, p. 012059, 2021.

[6] W. Xu, Y. Wang, and J. Zhao, “A hybrid modelling method for time series forecasting based on a linear regression model and deep learning,” *Applied Intelligence*, vol. 49, no. 7, pp. 2875–2888, 2019.

[7] R. Siddiqui, A. Khan, and M. Ahmed, “A Hybrid Demand Forecasting Model for Greater Forecasting Accuracy: The Case of the Pharmaceutical Industry,” *Supply Chain Forum: An International Journal*, vol. 22, no. 3, pp. 1–13, 2021.

[8] R. Rathipriya, M. Saranya, and K. Ramkumar, “Demand Forecasting Model for Time-Series Pharmaceutical Data Using Neural Networks,” *Neural Computing and Applications*, vol. 35, pp. 1945–1957, 2022.

Appendix

The following pseudocode outlines the sample selection process used to ensure the quality and relevance of the dataset for modeling tasks:

Algorithm 1 Sample Selection Algorithm

```
Require: Cleaned data $df$, configuration thresholds config
Ensure: Filtered dataset $final\_df$
$final\_df \leftarrow \emptyset$
for each unique combination of drug name and manufacturer $(d, m)$ in $df$ do
    $group\_data \leftarrow$ subset of $df$ for $(d, m)$
    if length of $group\_data <$ config.min_months or sum of consumption $= 0$ then
        Skip group                                ▷ Insufficient or sparse data
    end if
    $non\_zero\_ratio \leftarrow$ proportion of non-zero consumption in $group\_data$
    if $non\_zero\_ratio <$ config.sparsity_threshold then
        Skip group                                ▷ Data too sparse
    end if
    $acf\_values \leftarrow$ autocorrelation function of consumption in $group\_data$
    if $\max(acf\_values[1:]) <$ config.min_acf_threshold then
        Skip group                                ▷ Insufficient autocorrelation
    end if
    if no start date $\geq$ config.min_start_date in $group\_data$ then
        Skip group                                ▷ No recent data
    end if
    $variance \leftarrow$ variance of consumption in $group\_data$
    if $variance <$ config.min_variance_threshold then
        Skip group                                ▷ Variance too low
    end if
    $missing\_ratio \leftarrow$ maximum missing ratio for features in $group\_data$
    if $missing\_ratio >$ config.max_missing_ratio then
        Skip group                                ▷ Feature missing data too high
    end if
    $skewness \leftarrow$ skewness of consumption in $group\_data$
    if $|skewness| >$ config.max_skewness then
        Skip group                                ▷ Target variable too skewed
    end if
    $correlation \leftarrow$ correlation of consumption with lagged features in $group\_data$
    if $|correlation| <$ config.min_correlation then
        Skip group                                ▷ Insufficient correlation with features
    end if
    $final\_df \leftarrow final\_df \cup group\_data$
end for
return $final\_df$
```